AMATH 482/582: HOMEWORK 1

KATIE HIPPE

AMATH Department, University of Washington, Seattle, WA
kahippe@uw.edu

Abstract. A moving submarine in the Puget Sound emits an unknown acoustic frequency that is hidden within noisy acoustics pressure data. We have denoised this data using the Fourier Transform in order to reveal the true path of this submarine, and have graphed a surface-level representation to aid future tracking endeavors.

1. Introduction and Overview

Underneath the surface of the Puget Sound lurks a mysterious submarine, a new technology that emits an unknown acoustic frequency. This submarine completes a loop every 24 hours, though we don't know exactly where and how it moves. Luckily for us, we have obtained broad spectrum recording of acoustic pressure data across the submarine's known haunt over a 24 hour period, taken in half-hour increments. However, this pressure data is quite noisy, and we aren't sure which frequencies are from the submarine's emissions and which are simply extraneous. These frequencies are also detected on a three-dimension underwater grid, leading to complex data structures.

We wish to identify this submarine's path. Using Fourier Transforms, we will reveal the submarine's emission frequency. This will inform the filter we will use on our noisy pressure data to identify which frequencies actually correspond to the sub's positioning. This will reveal the submarine's daily route. From this 3-D path, we will also provide a 2-D simplification that ignores the submarine's changes in depth, so that we might follow it from above the surface.

2. Theoretical Background

Our initial noisy pressure data is given to us as 49 instances of a three-dimensional grid, where the value at each x, y, and z value is the frequency detected at that location. We utilized a three dimensional discrete Fast Fourier Transform (1) and its inverse (2):

$$\hat{f}(k_x, k_y, k_z) = \sum_{x_n=0}^{N-1} \sum_{y_n=0}^{N-1} \sum_{z_n=0}^{N-1} f(x_n, y_n, z_n) e^{\frac{-2\pi i}{N}(k_x x_n + k_y y_n + k_z z_n)} \tag{1}$$

$$f(x_n, y_n, z_n) = \frac{1}{N^3} \sum_{k_x=0}^{N-1} \sum_{k_y=0}^{N-1} \sum_{k_z=0}^{N-1} \hat{f}(k_x, k_y, k_z) e^{\frac{2\pi i}{N}(k_x x_n + k_y y_n + k_z z_n)} \tag{2}$$

where $f(x, y, z)$ is our discrete three dimensional data and N is the number of data points in the x, y, and z directions (here known to all be equal). The FFT transforms our signal data into the Fourier domain, where the value at each location now corresponds to the coefficient of that particular frequency. Thus, to find the maximum frequency, we need only find the location of the maximum value of the transformed data.

Next, to filter out the noise, we multiplied our transformed data by a Gaussian filter (3) (centered

Date: January 26, 2025.

around the location of the maximum frequency on the k-domain (k_x^*, k_y^*, k_z^*)) before taking the inverse Fourier Transform (2).

$$G(k_x, k_y, k_z) = e^{-\frac{1}{2\sigma^2}((k_x-k_x^*)^2+(k_y-k_y^*)^2+(k_z-k_z^*)^2)} \tag{3}$$

The result will be a reconstruction of our original data without extraneous frequency measurements that clutter the submarine's true path. From this three-dimensional data set, we may visualize the path on both a 3-D and top-down view.

3. Algorithm Implementation and Development

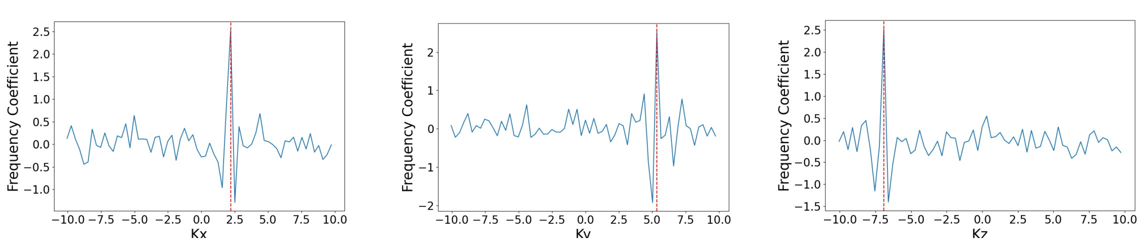
To develop and run these functions and algorithms, we used Python 3.12.8 in Visual Studios Code. For general numerical methods as well as the Fast Fourier Transform and its inverse, we used NumPy [1] . For plotting, we used Plotly [3] and Matplotlib [2].

First, we took the Fourier Transform of the noisy 3D data, then averaged it across all time steps. We found the location of the maximum frequency of this average, then constructed a three-dimensional Gaussian Filter to smooth away alternate frequencies, all within the Fourier domain. After we multiplied the transformed data (all time-steps, not the average) by the filter, we used the inverse Fourier Transform to get back our smoothed data. Finally, we found the location of the maximum frequency in the spatial domain at each time step to get a particular x, y, and z coordinate. Finally, we graphed the x and y coordinates over all time steps to get a two-dimensional top-down view of the submarine's path.

4. Computational Results

Our first major result was the location of the maximum frequency. We'd already transformed our noisy signal data into the Fourier domain, denoted here by Kz, Ky, and Kz axes, and averaged it across all time steps as our initial filtering technique. Now, the value at each (Kx, Ky, Kz) data point represented the coefficient value of a component frequency. We found the location of the maximum frequency within our data frame, here a 64x64x64 grid, then found the corresponding (Kx, Ky, Kz) value: $(k_x^*, k_y^*, k_z^*) = (2.199, 5.341, -6.112)$. The frequency coefficient at this location is $(2.527 + 1.108i)$.

To both visualize this result and support it as the true maximum, we graphed the real component of the frequency coefficient as a function of the three dimensions at this particular point, shown in Figure 1. Overlaid on each graph is a dotted red line that corresponds to (k_x^*, k_y^*, k_z^*). In each graph, we see a clear maximum that matches up with the calculated value. In the context of our situation, this means that the dominant frequency contributing to the acoustic frequency set off by our submarine is located at (k_x^*, k_y^*, k_z^*). While other minor frequencies contribute, this is the most important, and will be the most helpful to us as we attempt to identify the exact location of the submarine at each time step.

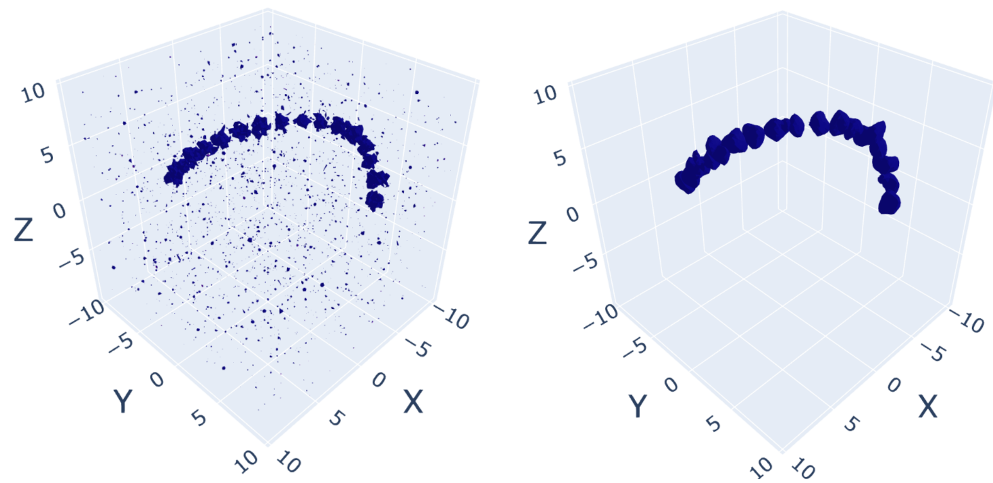


(A) Coefficients along Kx axis (B) Coefficients along Ky axis (C) Coefficients along Kz axis

Figure 1. 2-Dimensional slices of the location of maximum frequency with the x, y, and z locations overlaid.

Our next step was to create a Gaussian filter. We use $(k_x^*, k_y^*, k_z^*) = (2.199, 5.341, -6.112)$ in (3) to construct our Gaussian (G), then multiply our transformed data at each time step by G. Since G is centered around the maximum frequency, it will prioritize values in the Fourier domain that are close by and ignore values that are further away. Thus when we take the Inverse Fourier Transform, our data becomes smooth, such that extraneous frequencies not close to the maximum frequency have much decreased importance.

We may visualize this transformation in Figure 2, where we have graphed on our 3-D domain all the data at all time steps, normalized such that the maximum frequency is set equal to 1, both before and after the filtering process:



(A) Noisy Signal Data

(B) Cleaned Signal Data

FIGURE 2. Noisy and cleaned acoustic frequency data across all time steps across our three-dimensional domain.

The difference between the two images is striking. In making these graphs, we kept consistent a lower bound on the frequencies our graphed frequencies. However, this will not affect our future calculations with the denoised data as we are concerned with larger frequencies instead of smaller ones. We have used the same lower cut-off of .6 across both graphs.

Our noisy signals have higher frequencies spread out across the survey domain, shielding the path within. Although we can still identify a broad path by eye, this data is much more difficult to create a centralized, dependable prediction off of. Our cleaned signal data, on the other hand, has a much more cohesive path. The data points are crowded around it, making the path appear as a collection of thicker points. This is a result of the filtering process, as frequencies similar to the dominant frequency are prioritized. Now frequencies that may have been of lesser import in the noisy data have been boosted simply due to their vicinity to the dominant frequency. No matter what, we have narrowed down the potential area for the submarine to be in at each time step.

Overall, this denoising technique seems effective and valid, as our denoised path is consistent with what we would've expected by eye from our initial data. Any further filtering efforts should be focused on decreasing the range for the sub at each particular time step, or in other words, shrinking the size of the chunks in Figure 2b to yield an even smoother path. For our purposes, this cleaned data is sufficient.

Thus we may finally complete our initial objective. At each time step in the cleaned data, we isolated the maximum frequency value. Since our filter was centered around the dominant frequency, it was prioritized above all others, so the maximum value at each time step must correspond to it.

We then isolated the X and Y values corresponding to that maximum frequency, then graphed the two in Figure 3:

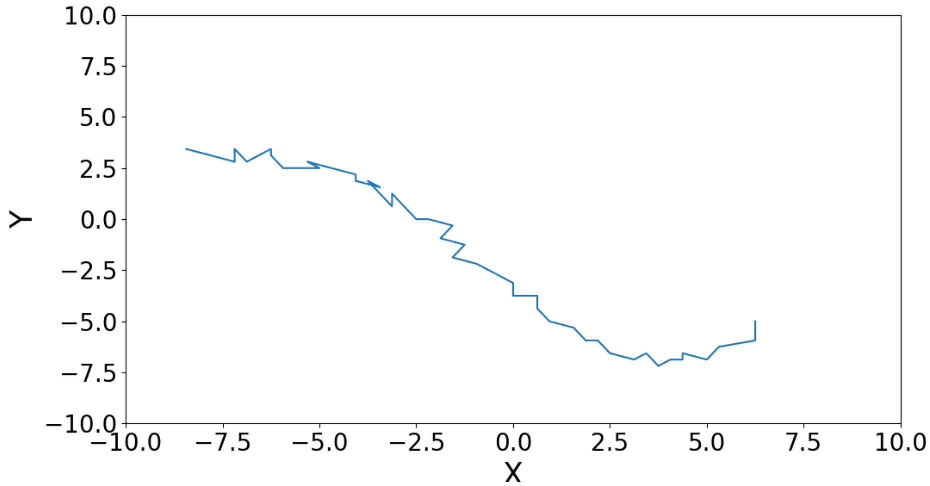


FIGURE 3. A top-down view of the submarine's path, with each data point connected based on time.

The submarine starts in the upper left-hand corner of the domain, then finishes its daily routine in the lower right-hand corner. The position at each time-step is connected to the next linearly based on sequential time step. Overall, the resulting path is quite smooth, though there are some hiccups in the path like switchbacks and jagged connections. This might be because of inadequate filtering, or a more complex path taken by the submarine that takes into account issues in the Z direction, such as avoiding obstacles or currents. We may now use this projected path to monitor the submarine in the future. With more data, we may be able to either confirm or extend the path suggested in this report.

This path will allow us to save effort and resources on the monitoring efforts so that we may focus our efforts on answering questions such as: why is this submarine lurking? What does it want with us?

5. Summary and Conclusions

Over the course of this project, we determined a probable reconstruction of the daily path of this mysterious submarine, and have provided a 2-D reconstruction with which to better monitor it. Using both a Gaussian and time-averaging filtered out much of the noise present in the original data set. In the future, we may try alternate filtering techniques in addition or instead of the ones attempted in this report to reveal an even smoother path of the submarine.

Acknowledgements

The author is thankful to Prof. Frank for discussion on the physical meaning of the Fourier transform over a 3D domain that includes a time-aspect. We are also thankful to our peers Kaillah Selvaretnam, Heidi Neuman, Kate Everling, and Laura Pong for conceptual help surrounding the problem and graphing our results, and especially Quinn Kelly for help debugging and formatting.

References

[1] C. R. Harris, K. J. Millman, S. J. der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, SebastianBerg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.

[2] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[3] P. T. Inc. Collaborative data science, 2015.